(12) UK Patent Application (19) GB (11) 2 328 593 (13) A

(43) Date of A Publication 24.02.1999

(21) Application No 9812553.7

(22) Date of Filing 11.06.1998

(30) Priority Data
(31) 97024145 (32) 11.06.1997 (33) KR

(51) INT CL[6]
H04L 12/56 // H04Q 11/04

(52) UK CL (Edition Q)
H4P PPS
H4K KTK

(56) Documents Cited
EP 0706298 A2 EP 0681385 A2

(58) Field of Search
UK CL (Edition P) H4K KTK , H4P PPS
INT CL[6] H04L 12/56
ONLINE DATABASE: WPI

(71) Applicant(s)
Samsung Electronics Co Limited
(Incorporated in the Republic of Korea)
416 Maetan-dong, Paldal-gu, Suwon-city,
Kyungki-do, Republic of Korea

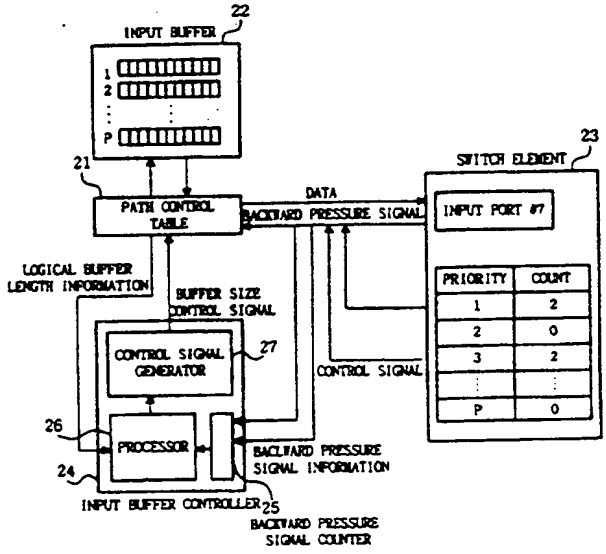
(72) Inventor(s)
Jeong Won Heo
Sung-Hyuk Byun
Ju Yong Lee
Jin Woo Yang
Jae Jin Shin
Kyung Guen Lee
Dan Keun Sung

(74) Agent and/or Address for Service
Dibb Lupton Alsop
Fountain Precinct, Balm Green, SHEFFIELD, S1 1RZ,
United Kingdom

(54) Abstract Title
Controlling logical queue depth in an input buffer of an ATM switch

(57) An input butler 22 of an ATM switch has a number of logical queues each for storing cells of a different priority (1,2....P). The logical queue depth (fill level) of each queue varies according to traffic conditions and incoming cells of a particular priority may be lost if there is no free space in the logical queue for that priority. To reduce the amount of cell loss the allocation of buffer space to each logical queue (logical queue size) is dynamically controlled according to a back pressure signal from the switch element 23 and the current depth being occupied in that logical queue. If the occurrence rate of a backpressure signal reaches a threshold and the logical queue depth reaches a threshold for a particular queue then an algorithm checks to see it there is unused buffer space in one of the other logical queues. If so this space is reallocated to the overfull logical queue.

FIG. 3



GB 2 328 593 A

FIG. 1

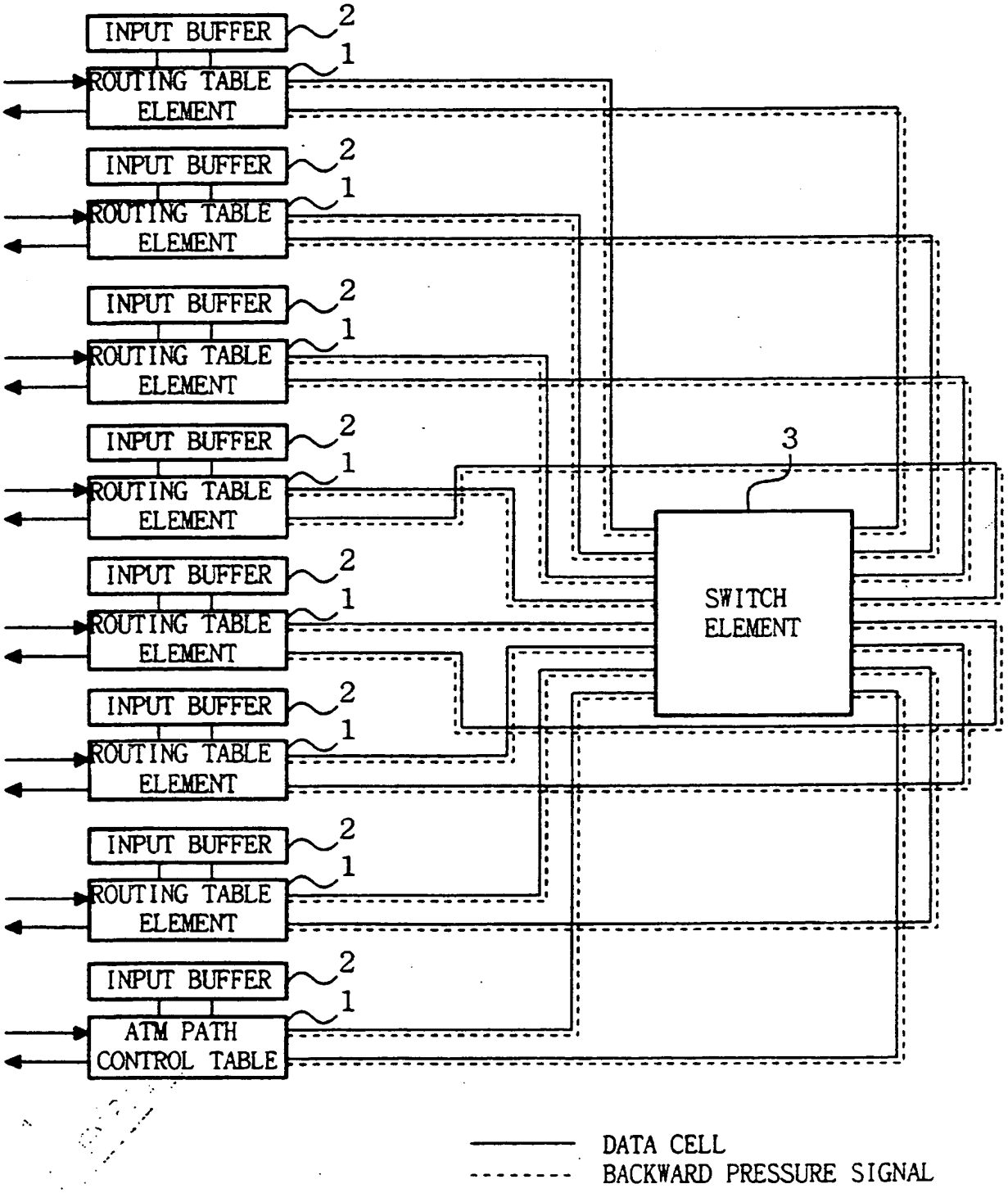

INPUT BUFFER
2
ROUTING TABLE ELEMENT
1
INPUT BUFFER
2
ROUTING TABLE ELEMENT
1
INPUT BUFFER
2
ROUTING TABLE ELEMENT
1
INPUT BUFFER
2
ROUTING TABLE ELEMENT
1
3
SWITCH ELEMENT
INPUT BUFFER
2
ROUTING TABLE ELEMENT
1
INPUT BUFFER
2
ROUTING TABLE ELEMENT
1
INPUT BUFFER
2
ROUTING TABLE ELEMENT
1
INPUT BUFFER
2
ATM PATH CONTROL TABLE
1
DATA CELL
BACKWARD PRESSURE SIGNAL

FIG. 2

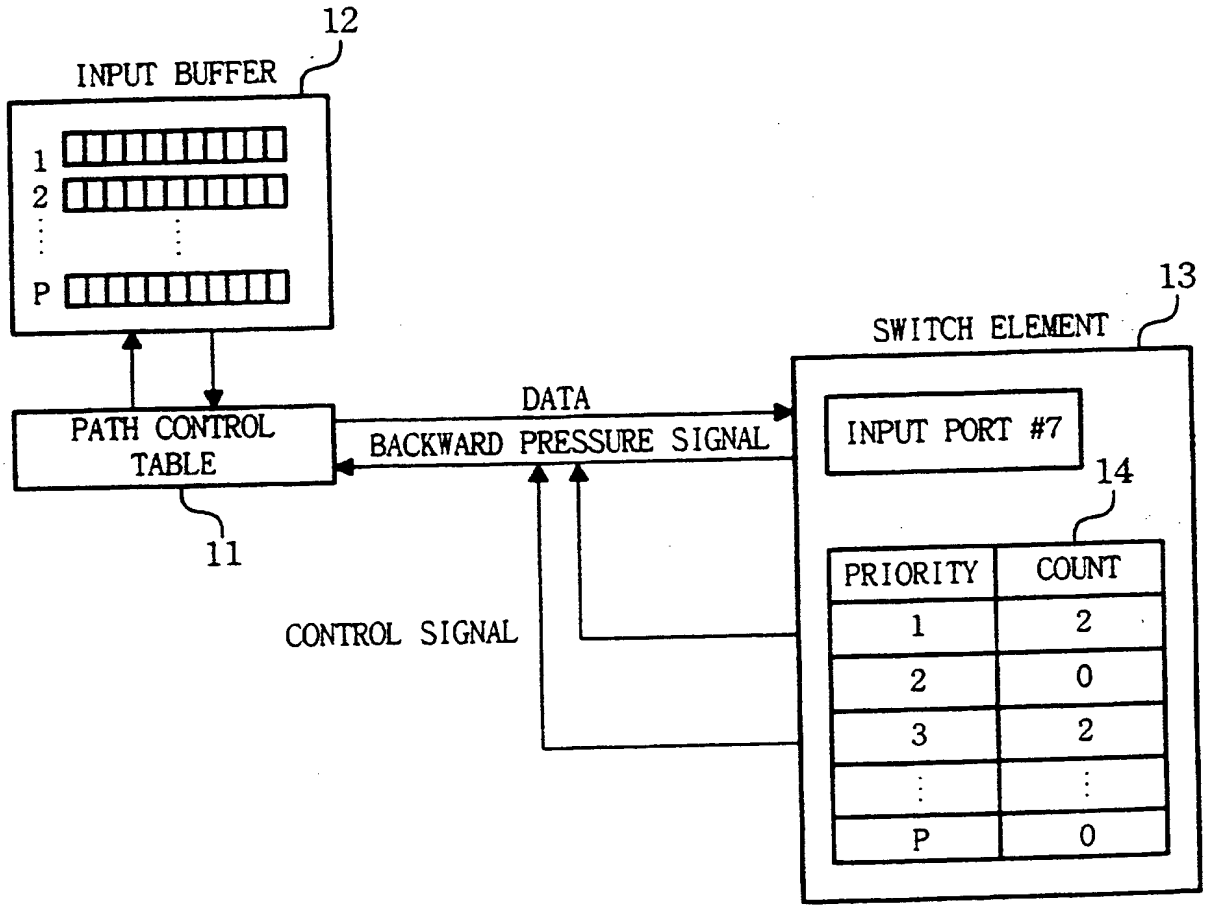

12
INPUT BUFFER
1
2
P
13
SWITCH ELEMENT
PATH CONTROL TABLE
DATA
BACKWARD PRESSURE SIGNAL
INPUT PORT #7
14
11
CONTROL SIGNAL
PRIORITY
COUNT
1
2
2
0
3
2
P
0

FIG. 3

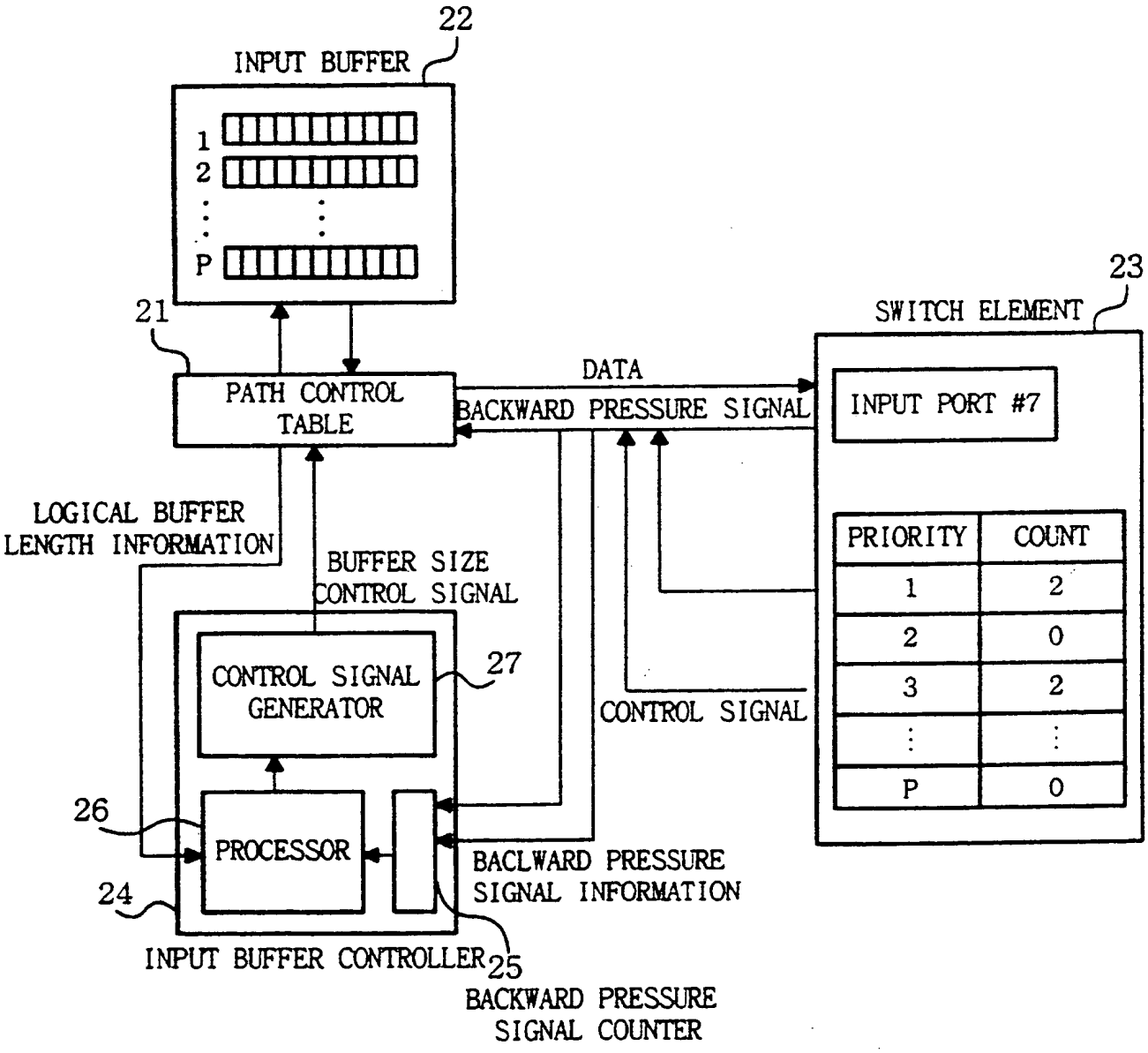

22
INPUT BUFFER
1
2
P
21
PATH CONTROL TABLE
DATA
BACKWARD PRESSURE SIGNAL
23
SWITCH ELEMENT
INPUT PORT #7
PRIORITY
COUNT
1
2
2
0
3
2
P
0
LOGICAL BUFFER LENGTH INFORMATION
BUFFER SIZE CONTROL SIGNAL
CONTROL SIGNAL GENERATOR
27
CONTROL SIGNAL
26
PROCESSOR
BACLWARD PRESSURE SIGNAL INFORMATION
24
INPUT BUFFER CONTROLLER
25
BACKWARD PRESSURE SIGNAL COUNTER

FIG. 4

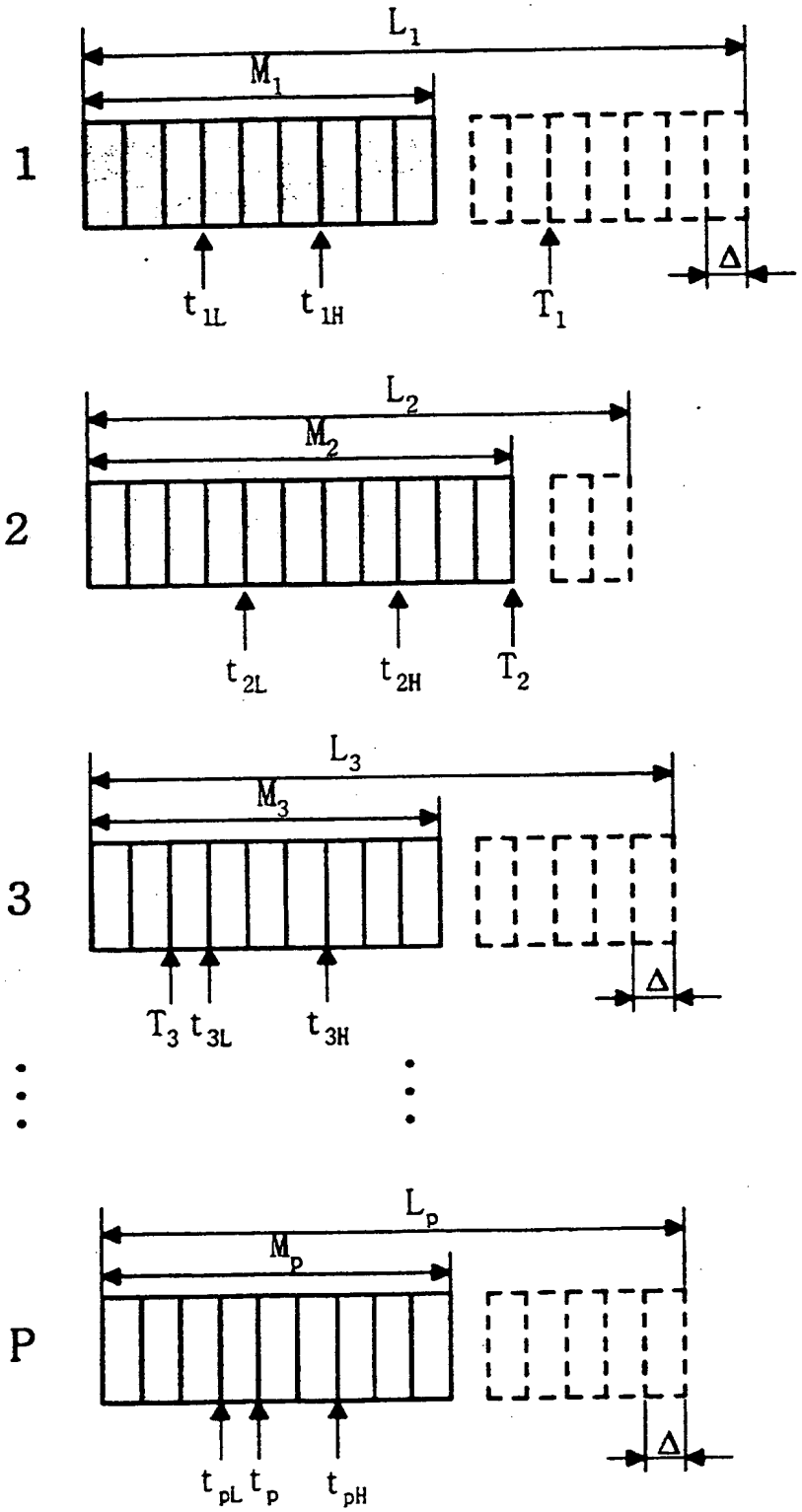
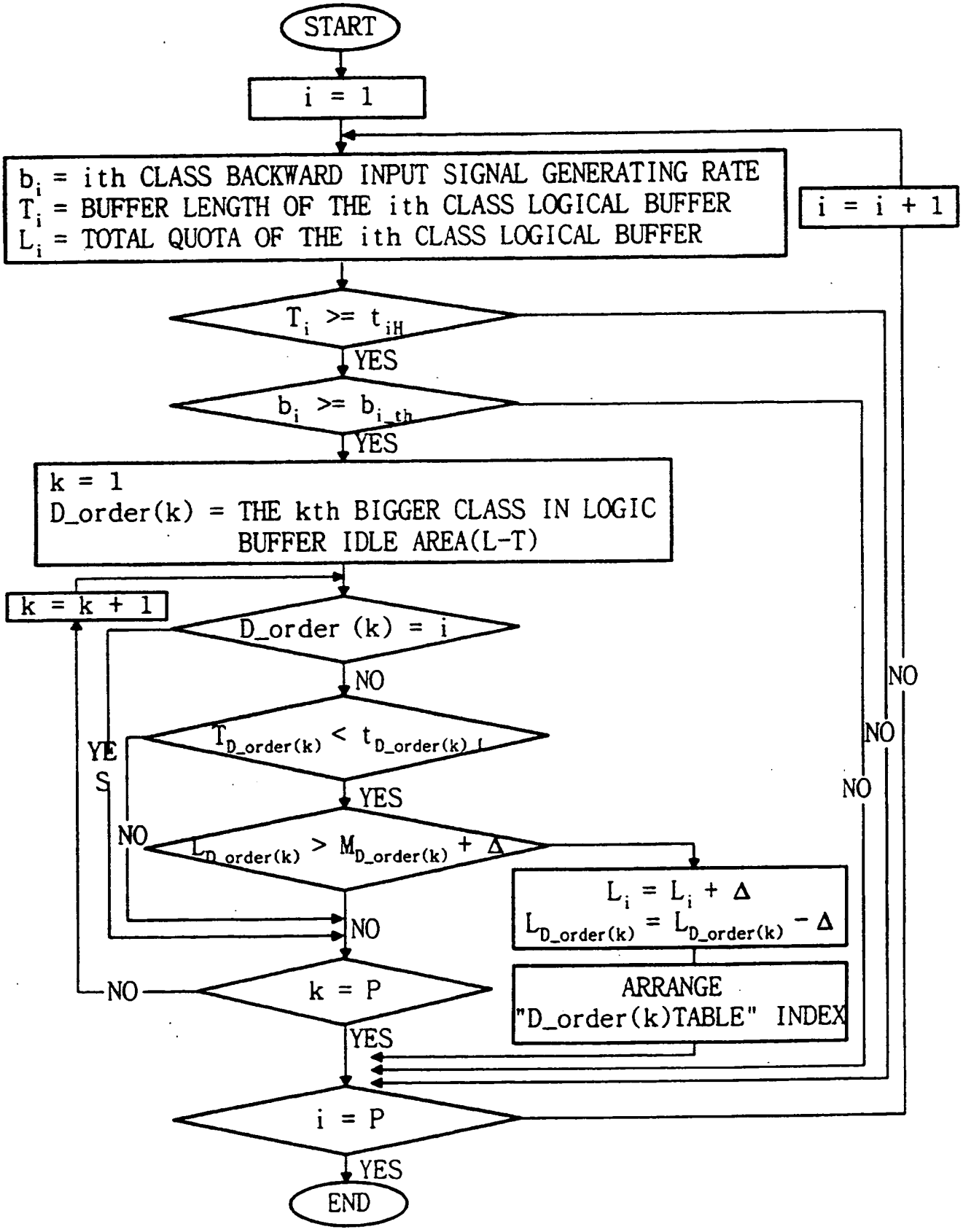

1
L_1
M_1
t_{1L}
t_{1H}
T_1
Δ
2
L_2
M_2
t_{2L}
t_{2H}
T_2
3
L_3
M_3
T_3
t_{3L}
t_{3H}
Δ
P
L_p
M_p
t_{pL}
t_p
t_{pH}
Δ

FIG. 5


START
i = 1
b_i = ith CLASS BACKWARD INPUT SIGNAL GENERATING RATE
T_i = BUFFER LENGTH OF THE ith CLASS LOGICAL BUFFER
L_i = TOTAL QUOTA OF THE ith CLASS LOGICAL BUFFER
i = i + 1
T_i >= t_iH
YES
NO
b_i >= b_i_th
YES
NO
k = 1
D_order(k) = THE kth BIGGER CLASS IN LOGIC BUFFER IDLE AREA(L-T)
k = k + 1
D_order (k) = i
NO
T_D_order(k) < t_D_order(k)_l
YES
NO
YES
L_D_order(k) > M_D_order(k) + Δ
NO
L_i = L_i + Δ
L_D_order(k) = L_D_order(k) - Δ
NO
k = P
NO
YES
ARRANGE "D_order(k)TABLE" INDEX
i = P
YES
END

APPARATUS AND METHOD FOR CONTROLLING LOGICAL QUEUE DEPTH

The present invention relates to an apparatus and method for controlling logical queue depth and more particularly to an input buffer controller of a switching system of the input buffering type that controls the cell-input to the switching system using a back-pressure signal.

Fig. 1 illustrates a system architecture of an NxN input buffer switching system using an input buffering method. The switching system includes a routing table element (1) for attaching a routing tag to an ATM input cell using the information of the output port, an input buffer (2) for storing the cells being input and a switching element (3) having a function of tranmitting cells between the input port and output port using the routing tag.

The routing table element (1) and the input buffer (2) are required on a one-to-one basis for every input port and the switching element may comprise a unit switch or several unit switches.

The cells input to the switching system are sent to a routing table element (1) which stores the input cell in an appropriate logical queue within the input buffer (2) according to the class of the input cell. In the switch system, an input buffer is divided to accommodate P logical queues which support the P priorities of input cells. The logical queue depth corresponding to each priority can be determined by the routing table element. A cell stored as

stated above is read from each logical FIFO queue according to the priority and transmitted to the routing table element (1). By using the output port information of the cell transmitted to the routing table element (1), the path information of a switch element (3) is determined and a tag is attached to the cell so that the path of the cell to be switched in said switch element is pre-determined. The cell including the tag attached by the routing table element (1) is transmitted to an output port through the switch element.

The method for transmitting a cell stored in one of the P logical queues to the routing table element (1) is such that it commences with the logical queue having the highest priority and progressively works through the queues according to progressively decreasing priority. If there is a cell to be transmitted in the logical queue that has been checked, a check is then made to determine if there is a back-pressure signal corresponding to the checked logical queue.

The switch element (3) has a table that stores the priority of the cell, which is in the shared buffer, and the number of each type of priority cell for each input port. If cells having the same priorities are transmitted to the same input port, a collision may occur between those cells. Therefore, a back-pressure signal is used to prevent the occurrence of such a situation.

In the case where the back-pressure signal does not exist, the cell is read from the input buffer (2) and transmitted to the switch element (3) through the routing

table element (1). In the case where a back-pressure signal exists, a check is made to determine if there is a cell stored for the logical queue of next priority and if there is a back-pressure signal corresponding to the next priority logical queue. Namely, a check is made to determine whether or not there is a cell in the logical queue and for the absence of a back-pressure signal, and then the cell is transmitted through the switch element (3) only if the two conditions are satisfied.

In Fig. 1, the solid line (4) between the switch element (3) and the routing table element (1) represents the transmission path of a data cell from the input buffer (2) to the switch element (3). The cell is transmitted to an output port of the switch element (3) via an input port of the switch element (3). The dotted line (5) represents the back-pressure signal that is transmitted from an input port to an output port side of the routing table element (1).

Fig. 2 illustrates the procedure for generation of the back-pressure signal explained briefly above. The unit switch element (13) stores an indication of the number of cells of particular priorities. The embodiment of Fig. 2 shows a table (14) illustrating the number of cells by priority that are stored in the input port 7. Suppose that the threshold value of each priority class is 2. There are two cells of the class with priority 1 and two cells of the class with priority 3. Accordingly, the switching element 13 transmits back-pressure signals to the routing table element (11) so that cells of the class with priority 1 and

cells of the class with priority 3 are not transmitted. The back-pressure signals are transmitted to the routing table element (11) through the corresponding input port when the number of cells of a class with a specific priority is greater than or equal to the threshold value by class, b, when the cells have been transmitted to a specific input port of the unit switch element (13) and when the total number of cells that are stored in the unit switch element (13) is greater than the pre-determined total threshold value, a. For example, in the case where the value of b is 2 and the value of a is 24, the back-pressure signals are transmitted to all of the input ports. In addition, when the number of cells of a specific class is two or more and the cells have been input to a specific input port of the unit switching element (13), the back-pressure signals are transmitted to the routing table in the front side through the corresponding specific input port.

In the method for controlling the input buffer by the input buffer controller in an input buffer switch as stated above, an input buffer can be used for the P logical queues to support the P priorities and the size of each logical queue can be determined in the routing table (11).

However, the static allocation of logical queue size has a disadvantage in that it cannot accept a dynamic traffic variation smoothly.

For example, in the case where cells having a number of priorities are input which have equal distribution and the same depths are allocated for each priority queue, if a

large number of cells with a specific priority or prioritues are input, empty areas may occur in the logical queue or queues having other priorities and a large number of Cell Losses may occur in the logical queue or queues having the specific priority or priorities as the buffers are insufficient. It can be appreciated that large cell losses can occur in systems having statically allocated buffer queue sizes.

It is an object of the present invention to mitigate at least some of the problems of the prior art.

Accordingly, the present invention provides an apparatus for controlling logical queue depth of a logical queue for storing an input cell in an ATM switch using a back pressure signal and information relating to the current utilisation of the logical queue, said apparatus comprising
a routing table element for making a tag for routing of an input cell,
an input buffer for storing the input cell in said logical queue,
a switch element for reading a cell from the input buffer and switching the input cell to an output port, and
an input buffer controller for controlling the logical queue size in the input buffer.

Preferably, an embodiment of the present invention provides a controller for the logical queue size in ATM switch system, using a back-pressure signal and occupied buffer depth information and supporting the P classes. The

controller includes Routing Table Element making tag for routing of input cell; Input Buffer storing the cell that a tag is attached to in the routing table element; Switch element that reads the cell from the input buffer and then switches it to the output port; and Input buffer controller controlling the logical queue size in the input buffer.

Advantageously, the present invention solves the problem as stated above using back-pressure signals in the ATM switch and an algorithm for determining the logical queue depth. The present invention also has the advantage of improving the Cell Loss Rate by effectively accepting the input cells having the P priorities using a input buffer, i.e., in order to make the logical queue depth for each priority dynamically variable, by dynamically allocating the priority buffer size that is allocated to each priority class using the information of logical queue depth such as the occurrence rate of the back-pressure signal, the allocated amount and the number of cells that are actually stored in the logical queue.

According to one embodiment of the present invention, it is preferable that the Cell Loss Rate is decreased by sending the logical queue size, which is determined periodically in the input buffer controller, to the Routing table element and by dynamically changing the logical queue size in the input buffer.

According to one embodiment of the present invention, it is preferable that the input buffer controller producing the logical queue size control signal further includes back-

pressure signal counter part for counting the number of generation of the back-pressure signal; processor calculating the logical queue allocation size using the information received from the back-pressure signal counter and the Routing Table Element; and control signal generating part producing the logical queue size control signal to send to the Routing Table Element.

According to one embodiment of the present invention, it is preferable that the Input buffer controller receives the information of back-pressure signal from the back-pressure signal counter part and receives the information of the buffer depth from the Routing Table Element, and then performs a calculation of the logical queue size using a method for determining of the logical queue size and generates the logical queue size control signal that is calculated from the control signal generating part.
In addition, the present invention discloses a method for determining of the

According to one embodiment of the present invention, it is preferable that the calculation of the logical queue size is performed in the processor.

According to one embodiment of the present invention, it is preferable that the logical queue size control signal is generated in the control signal generating part.

According to another aspect the present invention provides a method for controlling an input buffer. The method includes steps of calculating the back-pressure signal occurrence Rate bi of the ith class; calculating the

back-pressure signal occurrence threshold Rate bi_th of the ith class; calculating the buffer depth Ti of the logical queue of the ith class; calculating threshold values TiH, TiL of the two buffer depths of the ith class; calculating the buffer size Li of the logical queue of the ith class; calculating the empty area size Dj (j = 1, 2, 3,...,P) of logical queues for the number of p classes.

According to one embodiment of the present invention, it is preferable that the processor calculates the sizes of the P logical queues (Li, i = 1, 2, 3,...,P) according to the P classes within the input buffer every time interval W that it is a time interval for conversion of the logical queue size.

According to one embodiment of the present invention, it is preferable that for the purpose of calculating the size of the logical queue, the method further includes steps of: checking if the logical queue size Li can be changed or not; checking the logical queue of the next class if the logical queue size Li can not be changed; checking if the other logical queue size can be reduced to extend the size of the logical queue of the ith class to the extent of Δ; checking if the logical queue size ban be reduced in the descending order of the empty area of the logical queue, Lj-Tj for the number of P-1 of logical queues except a logical queue of the ith class; checking, beginning with the logical queue of the class with the grade of "D_order(1)" such that the logical queue has the largest empty area, if the buffer depth ($T_{D\_order(1)}$) is less than or equal to the threshold of

the buffer depth ($T_{D\_order(1),L}$) and if the additional allocation area ($L_{D\_order(1)}-M_{D\_order(1)}$) is greater than the minimum allocation (Δ); if two conditions are satisfied, reducing the logical queue depth ($L_{D\_order(1)}$) of the class with the grade of "D_order(1)" to the extent of Δ and extending the logical queue depth (Li) of the ith class to the extent of Δ; if all of two conditions are not satisfied, continuously checking the logical queue of the next class with changing the value of k; terminating the checking when a class satisfying the all of two conditions is found or when it has been checked up to the class with the grade of "D_order(1)" such that it is a logical queue having a minimum empty area; sorting and updating the data lists in the table storing the "D_order(k)" information after modifying the logical queue size; and performing the same operation for the logical queue of the i+1th class after determining the logical queue size of the ith class

An embodiment of the present invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Fig.1 shows an architecture of an NxN switching system;

Fig.2 shows an algorithm for generation of a back-pressure signal;

Fig.3 shows a structure illustrating an input buffer controller according to an embodiment;

Fig.4 shows a detailed view of logical queue for explaining an algorithm for determining the logical queue

depth according to an embodiment;

Fig.5 shows a flow chart illustrating an algorithm for determining the logical queue depth according to an embodiment.

Referring to figure 3, there is shown an input buffer controller comprising a back-pressure signal counter part (25) for receiving a back-pressure signal from a switch element (23) according to a priority class and for checking the number of received back-pressure signals, a processor (26) for calculating the size of logical queue allocation every time interval for updating the logical queue size, a control signal generating part (27) for generating a buffer size control signal to be sent to a routing table element.

The NxN input buffer switching system comprises an input buffer for temporally storing a cell being input to the switching system, the input buffer controller for controlling the cell being input to the input buffer, a routing table element for attaching a routing tag to the input cell using the information of an output port and a switching element having the cell-transmission function that reads the input cell from the input buffer and transmits it to the output port side using the routing tag.

An Input Buffer Controller using back-pressure signals to improve the Cell Loss Rate in an ATM switch comprises a back-pressure signal counter part for counting the number of generated back-pressure signals, a processor for calculating the logical queue size every time interval (W) of logical queue size using the number of back-pressure signals

received from the back-pressure signal counter part and using the size of each priority buffer received from the routing table element, and a control signal generating part for generating the buffer size control signal using the converted logical queue size received from the processor.

In order to assign dynamically the logical queue depth to an input buffer according to the input traffic characteristics, the input buffer controller (24) further comprises means for determining the sizes of p priority logical queues in the input buffer every time interval (W) for updating logical queue size by using the information such as the back-pressure signal from the switching element, the quantity of allocation for a logical queue by class and the depth of the buffer being occupied.

Fig.3 illustrates a structure of an input buffer controller according to an embodiment. The input buffer controller (24) includes a back-pressure signal counter part (25) that receives the back-pressure signal from the switch element (23) by priority class and checks the number of back-pressure signals, a processor (26) that calculates the size of logical queue allocation every time interval (W) for updating logical queue size and a control signal generating part (27) that generates a buffer size control signal to be sent to the routing table element. The routing table element (21) changes the size of buffer in the input buffers (22) every time interval (W), using the buffer size control signal.

In the input buffer controller, the back-pressure

signal counter part (25) receives the back-pressure signal and counts the number of back-pressure signals according to each priority class and then stores it.

The stored information is transmitted to the processor 26 wherein the information is that of the number of back-pressure signals which were counted during the time interval (W) defined as the time interval for updating a logical queue size. After the transmission of the occurrence count information for the back-pressure signals, the back-pressure signal counter part is initialised so that the occurrence number of the back-pressure signals can be counted during the next time interval for updating logical queue size.

The processor (26) calculates the logical queue size every time interval (W) using the following algorithm for determining of logical queue size using the information such as the occurrence count for the back-pressure signals that are received from the back-pressure signal counter part 25, information relaing to the size for each logical queue received from the routing table element and the number of cells occupying the logical queue at the present time.

The algorithm for determining of logical queue size finds ki, the occurrence count for the back-pressure signals for the cells of the class having the ith priority during the time interval (W). Therefore, the occurrence rate of the back-pressure signals for the cells of the class having the ith priority is the value of ki divided by W, time interval. That is, when it is assumed that the occurrence rate of the back-pressure signals is bi, the value of bi

becomes such that bi = ki / W.

Fig. 4 illustrates a detailed view of a logical queue to explain the algorithm for the determination of logical queue size. Fig. 5 shows a flow chart illustrating the algorithm for determining logical queue size. The algorithm is explained with reference to Fig. 4 and Fig. 5 as follows.

An input buffer (22) is split into p logical queues to support cells having at least one of p priorities and the logical queue for the cell having the ith priority has a fixed allocation area to store Mi cells. The Mi for the fixed allocation area denotes a value that is assigned statically but Li denotes the ith logical queue size and updated every time interval (W).

The logical queue for the cell with the ith priority has two thresholds, tiH and tiL (tiL ≤ tiH).

The tiH is the value that is used when it is necessary to increase the logical queue depth and tiL is the value that is used when it is necessary to decrease the logical queue depth. Assuming that Ti is the queue depth of the cell currently occupying the logical queue for the cell having the ith priority and Δ is the minimum unit for buffer allocation, the processor (26) calculates the logical queue depth every time interval (W).

In the case where the occurrence rate, bi, of the back-pressure signal for the cell with the ith priority is greater than the predetermined threshold, bi_th, and the queue depth, Ti, is greater than the threshold of the queue depth, TiH, in order to increase the logical queue depth for

the cell with ith priority by an amount Δ, the depth of the logical queue having the greatest difference between Lj and Tj(Lj-Tj) is decreased and its resources are allocated to the logical queue for the cell having the ith priority. Therefore, the depth of the ith logical queue is increased to Li+Δ from Li and the depth of the jth logical queue decreased by an amount Δ to Lj-Δ.

In the flow chart of Fig. 5, bi represents the occurrence rate for the back-pressure signal of the cell having the ith priority, Ti represents the number of cells currently being stored in the ith logical queue and Li represents the total size of the ith logical queue. The processor (26) calculates each logical queue depth Li (i=1,2,...,P) according to p priorities within the input buffer every time interval (W) for updating logical queue size, by using the algorithm for determining the logical queue depth as stated above. First, the processor checks whether the occurrence rate for the back-pressure signal of the cell having the ith priority bi is greater than the threshold, bi_th of the occurrence rate for the back-pressure signal and whether the number of cells being stored in the buffer, Ti is greater than the threshold, TiH of the buffer depth. If the two thresholds have been exceeded, the processor checks whether the logical queue size (Li) can be changed. If the two thresholds have not been exceeded, the processor checks the logical queue with the next priority class.

As the size of the logical queue of the other priority

class needs to be decreased by an amount Δ to increase the size of the specific logical queue by Δ, first of all, a determination is made as to whether the sizes of other logical queues could be decreased in descending order of logical queue empty area Lj-Tj (j=i) for the remaining p-1 logical queues except for the logical queue of ith priority class. In the above flow chart, "D_order (k)" (k=1,2,…,P) represents the class of the logical queue wherein the logical queue empty area is the kth largest. Namely, "D_order (2)" is regarded as the logical queue of the second priority class wherein the logical queue empty area is the second largest of the total p logical queues. A check is made as to whether the logical queue having the largest empty area of the "D_order (1)" class is less than or equal to the threshold of the buffer depth $t_{D\_order(1),L}$ and by checking whether the additional allocation area, $L_{D\_order(1)} - M_{D\_order(1)}$ is greater than the minimum allocation Δ. If the above two conditions are all satisfied, $L_{D\_order(1)}$, the logical queue size of the "D_order (1)" priority class, is decreased by an amount Δ and the logical queue size of the ith priority class is increased by an amount Δ. If any one of the two conditions is not satisfied, the checking of logical queue of the next priority class is continued by varying the value of k. If the priority class satisfying the two conditions is found or if it has checked to the "D_order (P)" priority class of the buffer having the smallest empty area, then processing terminates. The size of logical queue empty area is changed every logical queue size assignment.

Therefore, once the logical queue size is modified, the records of the table including the "D_order (K)" information is updated using sorting.

After determination of the logical queue size of the ith priority class as stated above, the logical queue size of the (i+1)th priority class is determined through the same steps. Here, the size of time interval (W), the size of the fixed allocation area Mi, two thresholds tiH and tiL, minimum unit for buffer allocation Δ and threshold of back-pressure signal ti_th etc. are determined according to the characteristics of switching system, input buffer size, the number of priority class and the input process of input traffic.

The control signal generating part (27) generates the logical queue size control signal every time interval (W) and transmits it to the routing table element (21). The input buffer controller (24) determines the logical queue size for each priority class and transmits it to the routing table element (21). The routing table element changes the logical queue size for each priority class dynamically and transmits the information of logical queue size for each updated priority class to the processor (26) in the input buffer controller (24). When a method for controlling the input buffer according to the present invention is used, the Cell Loss Rate can be improved. The input buffer controller (24) can comprise the extra controllers by each input port of a switch element or comprise a input buffer controller for controlling the all of the input buffers (22).

Fig. 3 illustrates an embodiment using a controller for each input port.

The input buffer controller (24) using a input buffering method advantageously controls the cells being input to the switching element (23) by using the back-pressure signal from the switching element (23) and the number of cells currently being stored in the logical queue, and it can change the logical queue size for each priority class dynamically and move to the cell input process dynamically to improve the Cell Loss Rate.

CLAIMS

1. Apparatus for controlling logical queue depth of a logical queue for storing an input cell in an ATM switch using a back pressure signal and information relating to the current utilisation of the logical queue, said apparatus comprising

a routing table element for making a tag for routing of an input cell,

an input buffer for storing the input cell in said logical queue,

a switch element for reading a cell from the input buffer and switching the input cell to an output port, and

an input buffer controller for controlling the logical queue size in the input buffer.

2. Apparatus as claimed in claim 1, wherein the input buffer controller comprises means for sending a logical queue size control signal to the routing table element to vary dynamically the logical queue size in the input buffer.

3. Apparatus as claimed in claim 3, wherein the input

buffer controller further comprises a back-pressure signal counter part for counting the number of occurrences of the back pressure signal;

a processor for calculating a logical queue size using information received from the back-pressure signal counter and the routing table element; and

a control signal generating part for producing the logical queue size control signal for output to the routing table element;

4. Apparatus as claimed in claim 3, wherein the buffer controller comprises means for receiving the information relating to the back-pressure signal from the back-pressure signal counter part, means for receiving the information from the routing table element, means for performing a calculation of logical queue size and means for generating the logical queue size control signal that is output from the control signal generating part.

5. Apparatus as claimed in claim 4, wherein the calculation of the logical queue size is performed by the processor.

6. Apparatus as claimed in either of claims 4 or 5, wherein the logical queue size control signal is generated in said control signal generating part.

7. Apparatus as claimed in any preceding claim, comprising means for calculating the back-pressure signal occurrence rate bi of the ith class;

means for calculating the back-pressure signal occurrence threshold rate bi_th of the ith class;

means for calculating the buffer depth Ti of the logical queue of the ith class;

means for calculating threshold values TiH, TiL of the two buffer depths of the ith class;

means for calculating the buffer size Li of the logical queue of the ith class;

means for calculating the empty area size Dj (j = 1, 2, 3,…,P) of logical queues for the number of p classes, wherein said processor calculates the sizes of said P logical queues (Li, i = 1, 2, 3, … ,P) according to the P classes within the input buffer every time interval (W).

8. Apparatus as claimed in claim 7, further comprising means for determining whether or not the logical queue size Li can be changed;

means for determining whether or not the logical queue size Lj of another logical queue can be changed;

means for determining whether the other logical queue size can be reduced to extend the size of the logical queue of the ith class to the extent of Δ;

means for checking if the logical queue size can be reduced in descending order of the empty area of the logical queue, Lj - Tj, for a remaining number, P-1, of logical queues except a logical queue of the ith class;

means for checking, beginning with the logical queue of the class with a grade of "D_order(1)" such that the logical queue has the largest empty area, if the buffer depth ($T_{D\_order(1)}$) is less than or equal to the threshold of the buffer depth ($T_{D\_order(1),L}$) and if the additional allocation area ($L_{D\_order(1)} - M_{D\_order(1)}$) is greater than the minimum allocation (Δ);

means, responsive to the two conditions being satisfied, for reducing the logical queue depth ($L_{D\_order(1)}$) of the class with the grade of "D_order(1)" to the extent of Δ and extending the logical queue depth (Li) of the ith class to the extent of Δ;

means, responsive to the two conditions not being satisfied, for checking the logical queue of the next class by changing the value of k;

means for terminating the checking when a class satisfying said all of two conditions is found or when the queue having a grade of "D_order(1)" such that it is a logical queue having a minimum empty area;

means for sorting and updating data lists in a table storing the "D_order(k)" information after modifying the logical queue size; and

means for performing the same operation for the logical queue of the i+1th class after determining the logical queue size of the ith class.

9. Apparatus substantially as described herein with reference to and/or as illustrated in the

accompanying drawings.

10. A method for controlling logical queue depth of a logical queue for storing an input cell in an ATM switch using a back pressure signal and information relating to the current utilisation of the logical queue, the method comprising the steps of calculating the back-pressure signal occurrence rate bi of the ith class;

calculating the back-pressure signal occurrence threshold rate bi_th of the ith class;

calculating the buffer depth Ti of the logical queue of the ith class;

calculating threshold values TiH, TiL of the two buffer depths of the ith class;

calculating the buffer size Li of the logical queue of the ith class;

calculating the empty area size Dj (j = 1, 2, 3,…,P) of logical queues for the number of p classes, wherein said processor calculates the sizes of said P logical queues (Li, i = 1, 2, 3, … ,P) according to the P classes within the

input buffer every time interval (W).

11. A method as claimed in claim 10, further comprising the steps of determining whether or not the logical queue size Li can be changed;

determining whether or not the logical queue size Lj of another logical queue can be changed;

determining whether the other logical queue size can be reduced to extend the size of the logical queue of the ith class to the extent of Δ;

checking if the logical queue size can be reduced in descending order of the empty area of the logical queue, Lj - Tj, for a remaining number, P-1, of logical queues except a logical queue of the ith class;

checking, beginning with the logical queue of the class with a grade of "D_order(1)" such that the logical queue has the largest empty area, if the buffer depth ($T_{D\_order(1)}$) is less than or equal to the threshold of the buffer depth ($T_{D\_order(1),L}$) and if the additional allocation area ($L_{D\_order(1)} - M_{D\_order(1)}$) is greater than the

minimum allocation (Δ);

reducing, responsive to the two conditions being satisfied, the logical queue depth ($L_{D\_order(1)}$) of the class with the grade of "D_order(1)" to the extent of Δ and extending the logical queue depth (Li) of the ith class to the extent of Δ;

checking, responsive to the two conditions not being satisfied, the logical queue of the next class by changing the value of k;

terminating the checking when a class satisfying said all of two conditions is found or when the queue having a grade of "D_order(1)" such that it is a logical queue having a minimum empty area;

sorting and updating data lists in a table storing the "D_order(k)" information after modifying the logical queue size; and

performing the same operation for the logical queue of the i+1th class after determining the logical queue size of the ith class.

12. A method substantially as described herein with reference to and/or as illustrated in the accompanying drawings.

The Patent Office

17

Application No: GB 9812553.7
Claims searched: All

Examiner: Gareth Griffiths
Date of search: 21 December 1998

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.P): H4K (KTK), H4P (PPS)

Int Cl (Ed.6): H04L 12/56

Other: Online Database: WPI

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A	EP0706298 A2 (AT&T)	
A	EP0681385 A2 (NEC)	

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

This Page Blank (uspto)